TP Analyse et visualisation des KPIs

Introduction au langage R

Aynaz ADL ZARRABI

aynaz.adlzarrabi@femto-st.fr

Novembre 2023

Table des matières

Plan détaillé

Introduction à R

R, Rstudio

C’est quoi les “dataframe” ?

Rmarkdown et R shiny

Lecture et écriture des fichiers externes à partir de R (Production du miel aux USA)

Types de données et objets

Les données d’exemple de production du miel aux USA

Extraction d’éléments

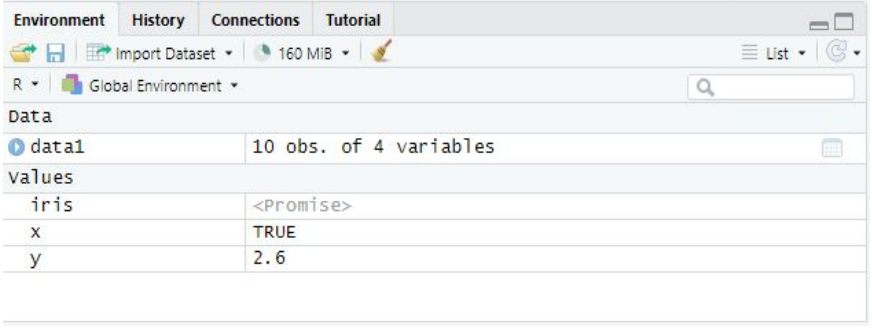
Conversion des types de données

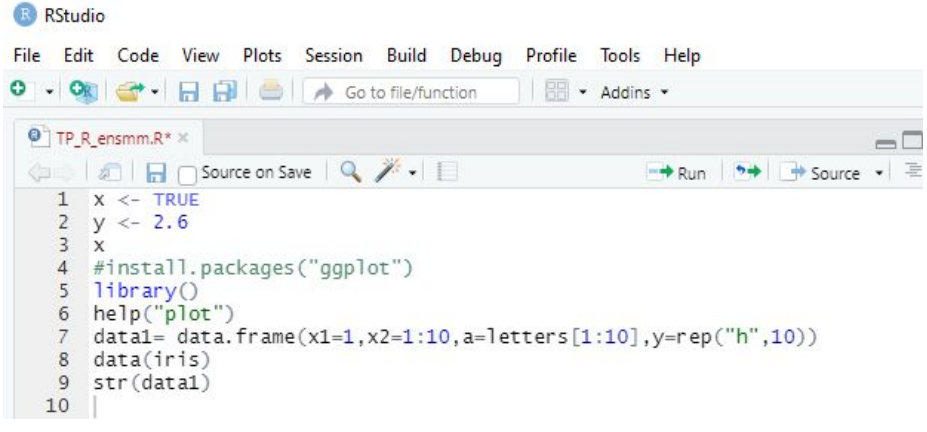
Introduction à R

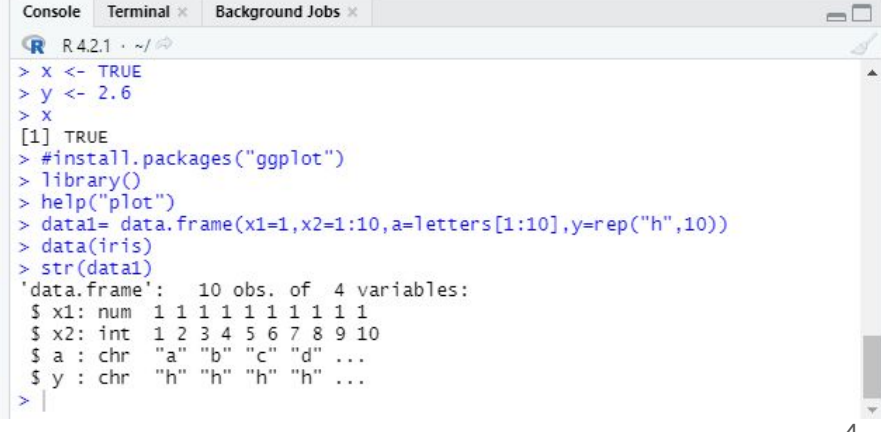
Le logiciel R est un freeware disponible sur le site

http://cran.r-project.org/

On peut utiliser un environnement de développement intégré (IDE) de R qu'on l'appelle **RStudio** sur ce site https://posit.co/download/rstudio-desktop/



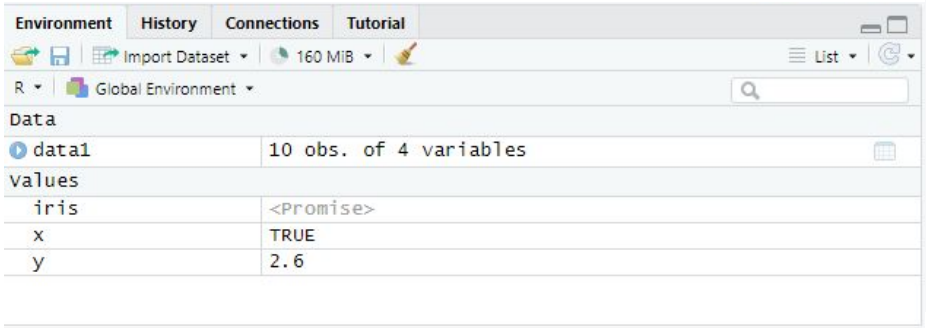


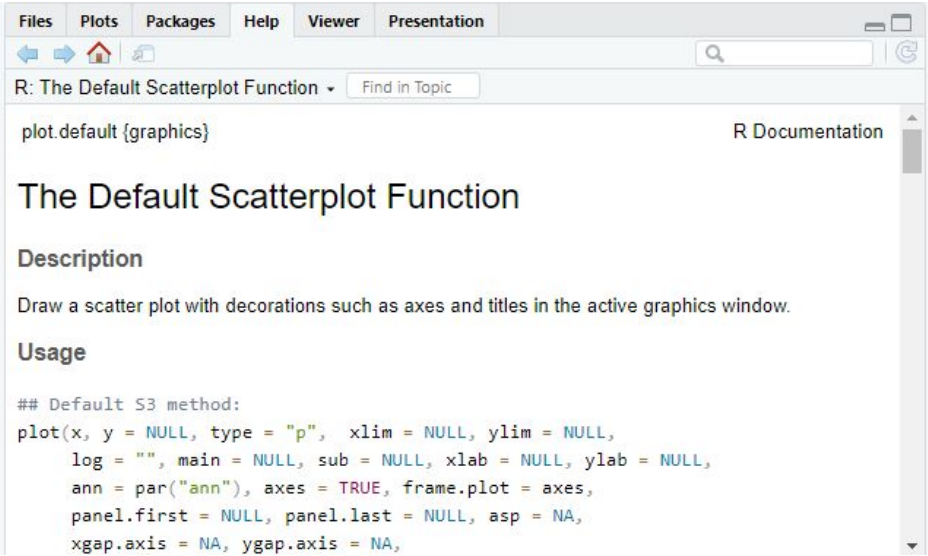


Introduction à R

Langage d'analyse statistique: qui est l'étude de la collecte de données, leur analyse, leur traitement, l'interprétation des résultats et leur présentation afin de rendre les données compréhensibles par tous.

Environnement de R





C'est quoi les "dataframe" ?

Il s'agit d'une structure spécifiquement conçue pour stocker des jeux de données, représenté sous la forme de tableaux de valeurs, stockés dans un fichier informatique

- **Données** : En statistique, des données sont des valeurs numériques (des nombres) ou alphanumériques (des chaînes de caractères), représentant les observations de certaines variables sur certains individus
- **Individu** ou unité statistique : Un individu est un élément de la population. L'ensemble des individus constitue la population
- **Variable** : Le terme variable désigne la représentation d'une caractéristique des individus

Creation de dataframe:

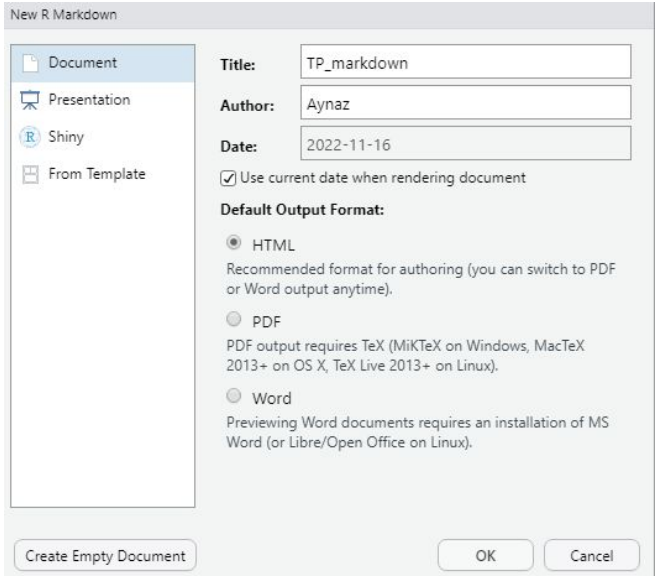
```
x1 = c(7, 3, 2, 9, 0)
x2 = c(4, 4, 1, 1, 8)
x3 = c(5, 3, 9, 2, 4)
Data1<-Data.frame (x1,x2,x3)
```

```
attributes(Data1)
names(Data1)
```

Rmarkdown

Markdown est une syntaxe de formatage simple pour la création de documents HTML, PDF et MS Word

```
install.packages("rmarkdown")
```




```
---
title: "TP_markdown"
author: "Aynaz"
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

```{r echo=FALSE}
summary(iris)
```
```

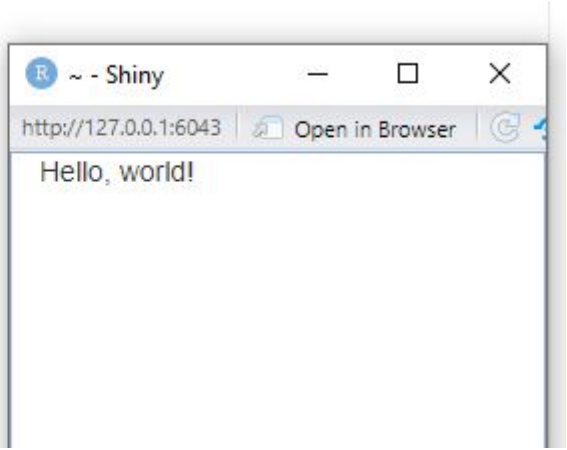
```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

R Shiny

Shiny est un package R qui permet de créer des applications Web interactives capables d'exécuter du code R sur le backend.

install.packages("shiny")

```
library(shiny)
ui <- fluidPage(
  "Hello, world!"
)
server <- function(input, output, session) {
}
shinyApp(ui, server)
```



Lecture et écriture des fichiers externes à partir de R

Répertoire de travail

Lorsqu'un fichier est identifié par un chemin d'accès relatif dans une commande, R doit faire une supposition quant au répertoire à partir duquel chercher le fichier,Pour connaître le répertoire de travail d'une session R:

```
getwd()
```

```
setwd("C:/coursR")
```

Lecture des fichiers csv

```
read.csv("data_ex.csv")
```

Lecture des fichiers excel

```
library(readxl)
```

```
read_excel("data_ex.xlsx")
```

Nos données d'exemple

```
new_data_frame<-as.data.frame(read.csv("https://www.data.gouv.fr/fr/datasets/r/63352e38-d353-4b54-bfd1-f1b3ee1cabd7", sep=";"))
```

Types de données et objets

Type de données

Integer(donnée entière), character(chaîne de caractères), logical (logique), double (données réelles)

as.(character/double/integer/logical)

Type d'objets

Vecteur, Matrice ("matrix","array"), Liste, Data Frame ("data.frame"), Facteur("factor")

Type des attributs

Type d'objet	names	dimnames	dim	class	levels
vecteur	(✓)	-	-	•	-
matrice ou array	-	(✓)	✓	•	-
liste	(✓)	-	-	•	-
data frame	(✓)	(✓, •)	•	✓	-
facteur	(✓)	-	-	✓	✓

Les données d'exemple (Prod du miel aux USA)

head(new_data_frame)

```
##    state numcol yieldpercol totalprod   stocks priceperlb prodvalue year
## 1     AL  16000          71    1136000   159000       0.72    818000 1998
## 2     AZ  55000          60    3300000  1485000       0.64   2112000 1998
## 3     AR  53000          65    3445000  1688000       0.59   2033000 1998
## 4     CA 450000          83   37350000 12326000       0.62  23157000 1998
## 5     CO  27000          72    1944000  1594000       0.70   1361000 1998
## 6     FL 230000          98   22540000  4508000       0.64  14426000 1998
## 7     GA  75000          56    4200000   307000       0.69   2898000 1998
```

str(new_data_frame)

```
## 'data.frame':    626 obs. of  9 variables:
##  $ state      : chr  "AL" "AZ" "AR" "CA" ...
##  $ numcol     : num  16000 55000 53000 450000 27000 230000 75000 8000 120000 9000 ...
##  $ yieldpercol: int  71 60 65 83 72 98 56 118 50 71 ...
##  $ totalprod  : num  1136000 3300000 3445000 37350000 1944000 ...
##  $ stocks     : num  159000 1485000 1688000 12326000 1594000 ...
##  $ priceperlb : num  0.72 0.64 0.59 0.62 0.7 0.64 0.69 0.77 0.65 1.19 ...
##  $ prodvalue  : num  818000 2112000 2033000 23157000 1361000 ...
##  $ year       : int  1998 1998 1998 1998 1998 1998 1998 1998 1998 1998 ...
##  $ consumption: num  977000 1815000 1757000 25024000 350000 ...
```

Q1. Qu'est-ce qu'on peut comprendre par le "summary" de notre dataframe?

Q2. C'est quoi la prochaine étape?

Extraction d'éléments

```
de <- c(2, 3, 4, 1, 2, 3, 5, 6, 5, 4)

names(de) <- c("I1", "I2", "I3", "I4", "I5", "I6", "I7", "I8", "I9", "I10")
```

Extraction d'un élément

```
de[1]   # I1
        # 2

de[[1]]  ## [1] 2
```

Plusieurs éléments

```
de[c(3, 6, 7)]
## I3 I6 I7
##  4  3  5

de[c("I3","I6","I7")]

colnames(new_data_frame), rownames(new_data_frame)
```

Traitement des dataframes

Combiner les différentes colonnes en une même dataframe

```
de1 <- c(2, 3, 4, 1, 2, 3, 5, 6, 5, 4)
de2 <- c(1, 4, 2, 3, 5, 4, 6, 2, 5, 3)
des <- cbind(de1, de2)
des2<- rbind(de1, de2)
```

Créer une nouveau dataframe à partir d'une autre dataframe

```
des_plus=data.frame(des, lanceur = rep(c("Luc", "Kim"), each = 5))
```

Extraction des éléments d'une dataframe

```
des_plus[c("de1", "de2")]
```

Facteurs (objet)

Factor w/ 2 levels

Conversion de type de données

Pour savoir avec quel type de données on travaille avec:

typeof(x) ,str(x)

Si on importe les données et on voit que l'importation n'est pas bien faite et on veut changer le types des variables :

as.numeric(), as.factor()

Si on a une liste et on veut la transformer en dataframe (et vice versa) tout en ajoutant les noms des variables (colonnes):

as.data.frame(), as.list(), *as.matrix()*, *as.vector()*

Résumé de Data frames

Objet récursif à 2 dimensions, conçu pour stocker :

- des jeux de données (ligne = observation, colonne = variable)
  - ≈ liste de vecteurs ou facteurs de même longueur (un élément = une colonne) ;
  - ≈ matrice avec données dont le type peut varier entre les colonnes ;
- fonctions de création :
  - data.frame, as.data.frame ;
- fonctions d'ajout de valeurs :
  - par concaténation avec : data.frame, cbind, rbind ;
- comment réaliser une extraction avec un opérateur :
  - [ou [[avec 1 argument identifiant une ou des colonnes ;
  - [ou [[avec 2 arguments, soit un par dimension ;
  - ou encore $ pour extraire une colonne nommée.

Plan détaillé

Prétraitement des données (nettoyage) en R

Variables numériques et qualitatives

Transformations de variables

Statistique descriptive de notre jeu de données

Manipulation de jeux de données

Variables quantitative (numériques) et qualitatives (catégorielles)

Variable numérique: discrète ou continue

Variable discrète:

• Si l'ensemble des valeurs que peut prendre la variable est fini ou dénombrable, la variable est discrète.

• pile ou face, nombre de voitures à un feu rouge, nombre d'actes dans un match de tennis..

Variable continue:

• Si l'ensemble des valeurs que peut prendre la variable est infini (R ou un intervalle de R) la variable est continue.

• Durée de trajet, taille, vitesse d'un service, longueur d'un saut...

Variable qualitative : réfère à une caractéristique qui n'est pas quantifiable. Une variable catégorique peut être nominale ou ordinale. Exp: Classement des élèves selon le comportement

Transformations des variables

- **Sélectionne des variables qui nous donne des informations utiles (qui ne sont pas nulles):**

```
new_data_frame <- subset(new_data_frame, select = c (dep,sexe,jour,hosp,rea,rad,dc))
```

- **Changements des noms des variables dans un dataframe:**

```
colnames(new_data_frame)<-c("department","gender","date","nb_hospitalizations","nb_reanimations","nb_returned_home","nb_deaths")
```

- **indiquer que la variable de sexe est une variable factorielle, car nous ne voulons pas traiter ses valeurs comme des valeurs numériques**

```
new_data_frame$gender<-factor(new_data_frame$gender,

labels=c("males & females","males","females"))

head(new_data_frame)
```

```
department          gender       date nb_hospitalizations
        01 males & females 2020-03-18                    2
        01           males 2020-03-18                    1
        01         females 2020-03-18                    1
        02 males & females 2020-03-18                   41
        02           males 2020-03-18                   19
        02         females 2020-03-18                   22
nb_reanimations nb_returned_home nb_deaths
              0                1         0
              0                1         0
              0                0         0
             10               18        11
              4               11         6
              6                7         5
```

Statistique descriptive de notre jeu de données

Fonction summary:

- vecteur numérique : minimum, premier quartile, médiane, moyenne, troisième quartile, maximum ;
- facteur : fréquences des modalités (comme la fonction table vue plus loin) ;
- matrice ou data frame : la fonction summary est appliquée séparément à chacune des colonnes.

summary(new_data_frame)

```
change <- honeyfinaldf %>%
  group_by(year) %>%
  summarise(YearTotal=sum(totalprod))
change
```

```
## # A tibble: 15 × 2
##     year YearTotal
##    <int>     <dbl>
##  1  1998 219519000
##  2  1999 202387000
##  3  2000 219558000
##  4  2001 185748000
##  5  2002 171265000
##  6  2003 181372000
##  7  2004 182729000
##  8  2005 173969000
##  9  2006 154238000
## 10  2007 147621000
## 11  2008 162972000
## 12  2009 145068000
## 13  2010 175294000
## 14  2011 147201000
## 15  2012 140907000
```

Manipulation des données de Covid with dyplyr

```
install.packages("dplyr"), library(dyplyr)

productionnper1998 <- honeyfinaldf %>%
  filter(year==1998) %>%            filtrer nos données avec une seule catégorie d'une variable qualitative comme l'année
  group_by(state) %>%               regrouper les données en fonction de la variable "state" (État)
  summarise(sumprod=sum(totalprod)) %>%   Filtrer les données et avoir un résumé statistique calcule la somme totale de la variable "totalprod" (production totale)
  arrange(desc(sumprod)) %>%        trie les données en ordre décroissant en fonction de la variable "sumprod",
  head(10) %>%
  mutate(percentage=round(sumprod/sum(sumprod)*100,2))  crée une nouvelle variable "percentage" qui calcule le pourcentage de la production totale de chaque État
```

Plan détaillé

Graph mining

Génération des plot avec le package de ggplot2

Les différences entre plots

Modifications des paramètres

Mettre en commun des jeux de données

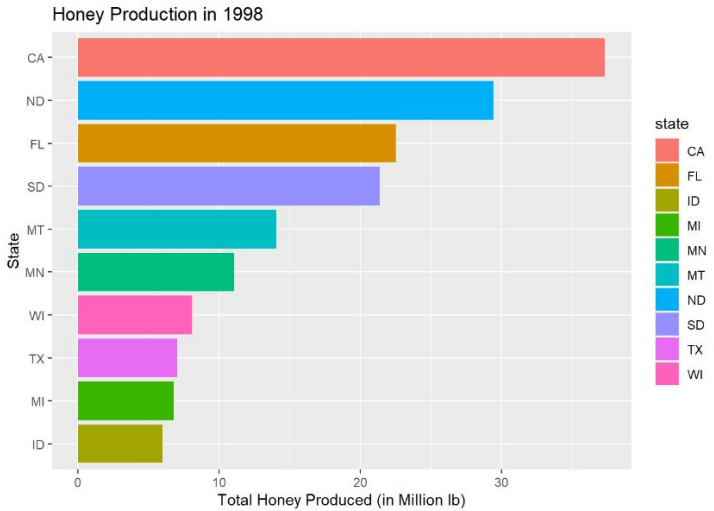
Génération des plot avec le package de ggplot2

We use the package *ggplot2* to map variables to nice graphics

```
install.packages("ggplot2"), library(ggplot2)
```

- graphique à barres pour représenter la production de miel totale par État aux États-Unis en 1998

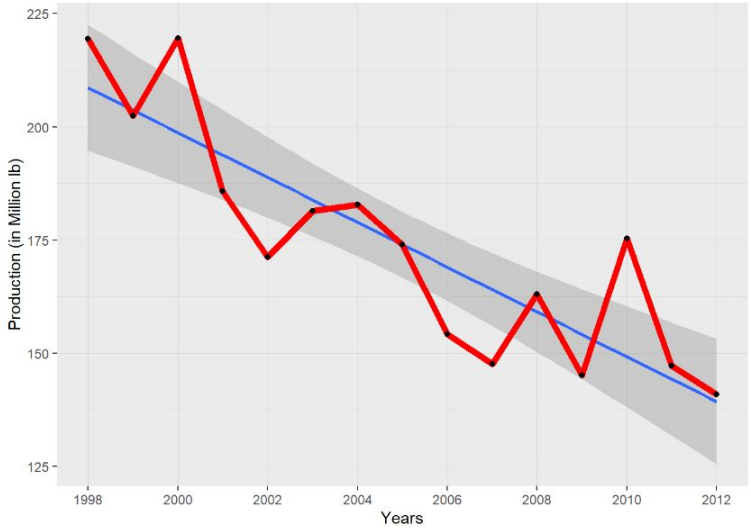
```
y1998 <- ggplot(data = productionnper1998, aes(x=reorder(state,sumprod),y=sumprod/1000000, fill = state)) +
 geom_bar(stat = "identity") +
 guides() +
 xlab("State") +
 ylab("Total Honey Produced (in Million lb)") +
 ggtitle("Honey Production in 1998") +
 coord_flip()
y1998
```



Les différences entre plots

- Lorsque vous souhaitez montrer la tendance au fil des ans, il est généralement plus approprié d'utiliser un line plot car il permet de visualiser clairement comment la variable évolue avec le temps. Un boxplot, en revanche, ne serait pas adapté pour montrer cette évolution temporelle, car il ne représente pas les données dans un ordre séquentiel.

```
ggplot(data=change, aes(x=year, y=YearTotal/1000000)) +
  geom_smooth(method = "lm") +
  geom_line(color = "red", size = 2) +
  geom_point() +
  scale_x_continuous(breaks=seq(1998, 2012, 2)) +
  ylab("Production (in Million lb)") +
  xlab("Years") +
    background_grid(major = "xy", minor="y", size.major = 0.2)
```



Plan détaillé

R shiny

Développement web en R

UI elements

Plot interactif

Développement web en R

Comment créer une interface utilisateur interactive et une logique serveur pour afficher les données et les analyses basées sur le pays sélectionné par l'utilisateur ?

```
install.packages("shiny"), library(shiny)

library(dplyr)

new_data_frame <- read.csv(file="covid_data.csv",header=T,sep=";")

new_data_frame <- new_data_frame %>% select(c(dep,sexe,jour,hosp,rea,rad,dc))

colnames(new_data_frame)<-c("department","gender","Number of hospitalizations","Number of reanimations","Number of returned home","Number of deaths","date")

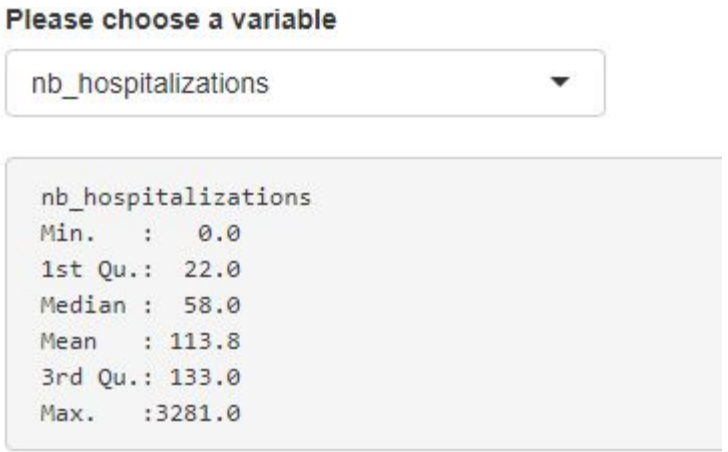
new_data_frame$gender<-factor(new_data_frame$gender,labels=c("males & females","males","females"))
```

UI elements

```
ui <- fluidPage(
 varSelectInput("variable", label = "Please choose a variable", new_data_frame),
 verbatimTextOutput("summary"),
 tableOutput("table")
 )
```

Plot interactif

```
server <- function(input, output, session) {
dataset <- reactive({new_data_frame  %>% select(c(!!input$variable))})
 output$summary <- renderPrint({ summary(dataset()) })
 output$table <- renderTable({head(dataset())})}
shinyApp(ui, server)
```



Résumé des fonctions

1. aggregate():est utilisée pour appliquer une fonction aux données subdivisées par un ou plusieurs facteurs (groupes).

```
moyenne_totalprod_par_an <- aggregate(totalprod ~ year, data = honey_data, FUN = mean)
```

2. apply() : s'applique à un tableau ou à une matrice pour exécuter une fonction sur les lignes ou lescolonnes

```
apply(honey_data[, c("yieldpercol", "priceperlb")], 2, mean, na.rm = TRUE)
```

3. arrange(): nous pouvons trier les données

```
donnees_triees <- arrange(honey_data, desc(totalprod))
```

4. summarise() et group_by() : sont souvent utilisées ensemble pour créer des résumés statistiques par groupe.

```
moyenne_numcol_par_etat <- honey_data %>%

                group_by(state) %>%

                summarise(moyenne_numcol = mean(numcol, na.rm = TRUE))
```

Résumé des fonctions

5. table(): créer un tableau de contingence des données.

```
tableau_annee <- table(honey_data$year)
```

6. theme(): permet de personnaliser l'apparence d'un graphique.

```
ggplot(moyenne_numcol_par_etat, aes(x = state, y = moyenne_numcol)) +
 geom_bar(stat = "identity") +
 theme(axis.text.x = element_text(angle = 90))
```

TP, le dataframe de Vin

Le dataframe wine_data contient des critiques de vins, incluant des évaluations (points), des informations sur le prix, le pays d'origine, la province, la région, le nom du dégustateur et le nom de la cave. Il offre un aperçu des caractéristiques et des perceptions de divers vins du monde entier, permettant des analyses approfondies sur la qualité, la popularité et la valeur des vins en fonction de divers facteurs géographiques et descriptifs.